### **Documentação do Projeto: Análise Preditiva de Risco de Cancro do Pulmão (Pneumologista João)**

*Arthur Trindade*

#### **1. Objetivo do Projeto**

O Dr. João, renomado pneumologista de João Pessoa, deseja otimizar sua prática médica utilizando análise de dados e inteligência artificial. O objetivo é desenvolver um modelo de Machine Learning capaz de identificar o risco de cancro do pulmão em pacientes, com base nos seus dados clínicos e de estilo de vida.

#### 

#### **2. Processo de ETL (Extração, Transformação e Carga)**

O ETL é a base do nosso projeto, onde preparamos os dados brutos para que se tornem informações úteis para análise e modelagem.

**2.1. Extração**

A primeira etapa consiste em carregar os dados dos pacientes a partir do arquivo pneumologista\_joao\_pacientes.csv.

* **Código:** df = pd.read\_csv("pneumologista\_joao\_pacientes.csv")
* **Explicação:** Utilizamos a biblioteca pandas para ler o arquivo CSV e carregá-lo para uma estrutura de tabela (DataFrame) chamada df, que nos permite manipular os dados em Python.

**2.2. Transformação**

Nesta fase, realizamos a limpeza, padronização e enriquecimento dos dados.

* **Tratamento de Dados Vazios (Nulos):** Identificamos que as colunas saturacao\_oxigenio, peso\_kg e altura\_m possuíam alguns valores em falta. Preenchemos os campos com a **mediana** da sua respectiva coluna. A mediana é uma medida estatística robusta, ideal para este cenário clínico, pois não é influenciada por valores extremos.
* **Padronização da Coluna sexo:** A coluna sexo continha várias representações para o mesmo género (ex: "F", "Fem", "Feminino"). Padronizámos estes valores para as categorias "Feminino", "Masculino" e "Outro" para garantir consistência.
* **Cálculo e Validação do IMC:** Criámos uma nova coluna imc\_calculado com a fórmula peso / (altura²). Isto não só nos dá uma métrica de saúde importante, mas também nos permite validar os valores da coluna imc original, garantindo a precisão dos dados.

**2.3. Carga**

Após a transformação, guardamos o conjunto de dados tratado.

* **Código:** df.to\_csv("pacientes\_tratado\_para\_analise.csv", index=False)
* **Explicação:** O DataFrame df, agora limpo, padronizado e com a coluna de IMC validada, é guardado num novo arquivo. Este arquivo está pronto para ser utilizado em análises clínicas ou para alimentar o nosso modelo de previsão.

#### **3. Análise com NumPy e Matplotlib**

Antes de construir o modelo, exploramos os dados para extrair informações clínicas relevantes.

* **Análise com NumPy:** Utilizamos a biblioteca NumPy para calcular a **média e o desvio padrão** do IMC de todos os pacientes. Estas métricas dão ao Dr. João uma visão geral e rápida do perfil corporal dos seus pacientes e da variabilidade entre eles.
* **Visualização com Matplotlib:**
  + **Gráfico de Barras:** Geramos um gráfico de barras para visualizar a quantidade de pacientes com e sem risco de cancro do pulmão. Isto ajuda a entender a prevalência do risco na base de dados.
  + **Gráfico de Linha:** Criámos um gráfico de linha que mostra a **média da saturação de oxigénio por faixa etária**. Este tipo de gráfico é excelente para identificar tendências, como por exemplo, se a saturação de oxigénio tende a diminuir com a idade.

#### **4. Construção do Modelo de Classificação**

Esta é a fase onde utilizamos Machine Learning para criar um sistema de apoio à decisão para o Dr. João.

1. **Preparação e Pré-processamento:**
   * Separamos os dados em X (as características do paciente que usamos para prever) e y (o que queremos prever: risco\_cancer\_pulmao). Colunas não preditivas como nome\_paciente são removidas.
   * Utilizamos um ColumnTransformer para aplicar tratamentos diferentes a colunas diferentes, convertendo todas as características para um formato numérico que o modelo consegue entender (StandardScaler para números e OneHotEncoder para texto).
2. **Divisão, Treino e Avaliação:**
   * **train\_test\_split**: Dividimos os dados em 80% para treinar o modelo e 20% para o testar, simulando uma situação real de previsão para novos pacientes.
   * **Treino (.fit)**: Treinamos um modelo RandomForestClassifier, que é muito eficaz em problemas de classificação complexos como este.
   * **Avaliação (accuracy\_score)**: Medimos a **acurácia** do modelo, que nos diz a percentagem de vezes que o modelo acertou o diagnóstico nos dados de teste.
3. **Guardar o Modelo Final:**
   * **Código:** joblib.dump(best\_model, "risco\_cancer\_model.pkl")
   * **Explicação:** O modelo treinado é guardado num ficheiro .pkl. Este ficheiro é a "inteligência" do projeto e pode ser integrado num software no consultório do Dr. João para fornecer uma segunda opinião baseada em dados para novos pacientes.